

Student Name:

Student id:

Sect #: Serial #:

University of Bahrain

Department of Computer Science

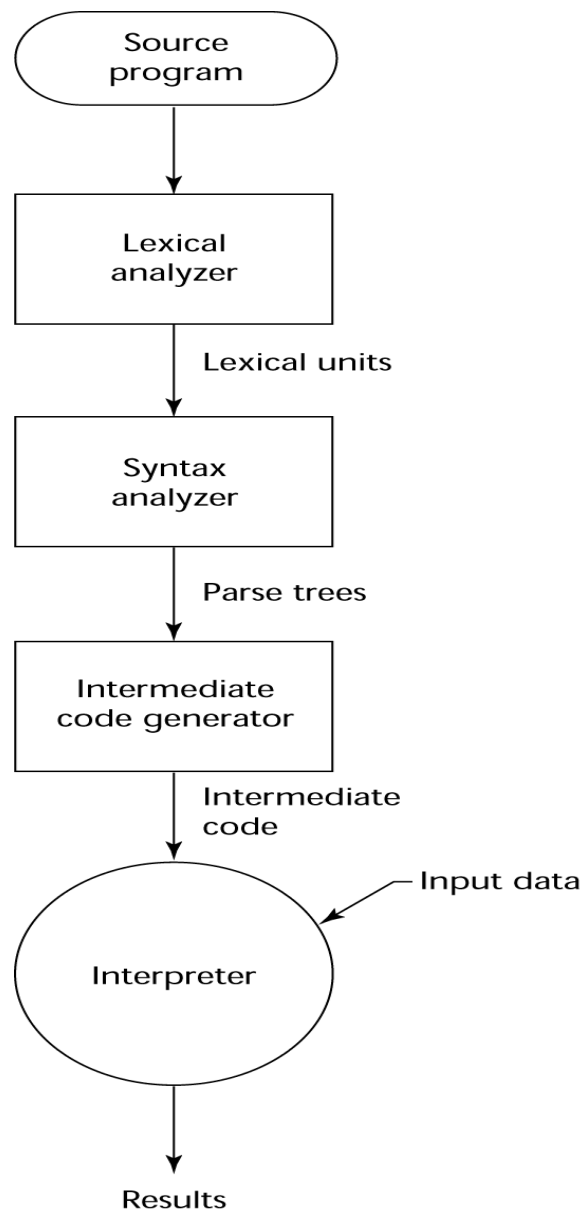
College of Information Technology

ITCS332: Organization of Programming Languages

Quiz #1: Chapter 1

Q1) Draw a flowchart that describes the phases of a hybrid implementation system.

[5 pts]



Q2) Briefly answer each of the following questions.

[6 pts]

- 1) Suggest two features of that improve the overall simplicity of a programming language:
 - **Reduce the number of language features and constructs**
 - **Use few feature multiplicity (means of doing the same operation)**
 - **Minimize the operator overloading.**
- 2) Suggest two language features to speed up the execution time of programs in a language:
 - **No checking of array indices against bounds.**
 - **Perform data type checking during compilation time.**
 - **Include more optimizations in the compiler.**
 - **Use preprocessor macros rather than subprograms (Functions).**
- 3) A language preprocessor increases the **compilation** times of programs and reduces the **execution** times of programs.
- 4) A semantics analyzer accepts as input **a parse tree** and produces as output **an intermediate code.**

Student Name:

Student id:

Sect #: Serial #:

University of Bahrain

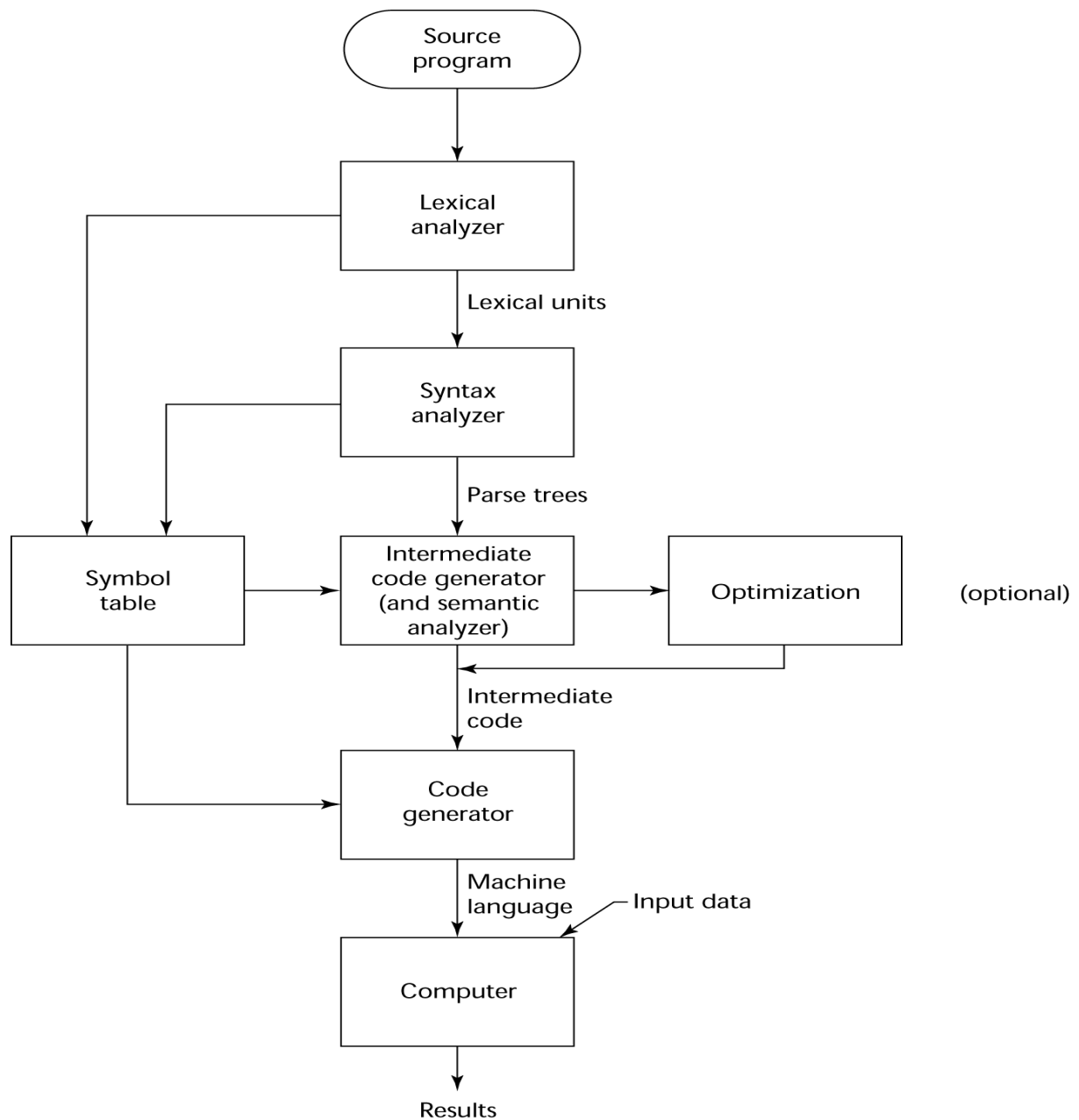
College of Information Technology
Department of Computer Science

ITCS332: Organization of Programming Languages

Quiz #1: Chapter 1

Q1) Draw a flowchart that describes the phases of a compiling implementation system.

[5 pts]



Q2) Briefly answer each of the following questions.

[6 pts]

5) Suggest two language features to speed up the compilation time of programs in a language:

- **Perform data type checking during execution time.**
- **Exclude preprocessor macros.**
- **Reduce (eliminate) the degree of optimizations in the compiler.**

6) Suggest two language features that improve the reliability of C++:

- **Perform checking of data types at compilation time.**
- **Include checking of array indices against bounds.**
- **Eliminate (reduce) aliasing.**
- **Exclude pointers from C++.**

7) The Syntax analyzer accepts as input **a sequence of lexemes** and produces as output **a parse tree**.

8) The main bottleneck of compiling systems is **CPU_Main memory speed gap** and the main bottleneck of interpreting systems is **statement decoding**.